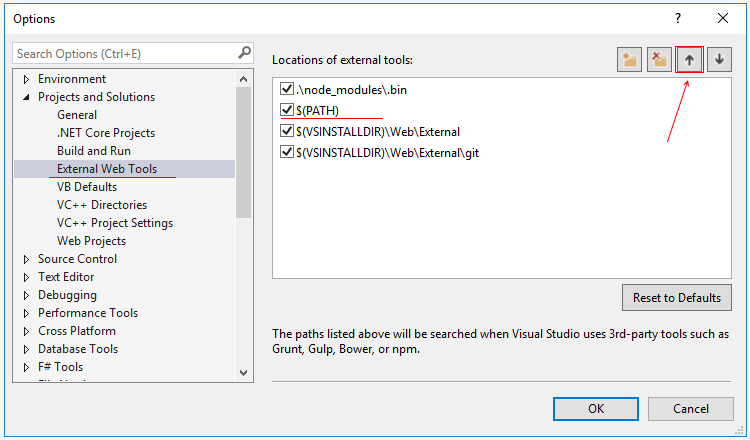
# Тестовое задание для Лаборатории Касперского. Редактор Книг

## Сборка проекта

### Требования

**Для сборки проекта необходимо**:

1. На компьютере должен быть установлен «**Node.JS**» и установщик пакетов «**npm**». Возможно необходимо установить еще и «**angular-cli**» – <https://nodejs.org/en/download/>;
2. Необходимо наличие установленной **Visual Studio 2015 Update 3** (или выше);
3. В **Visual Studio** необходимо сделать доп. настройку – в меню **Tools** -> **Options** на вкладке «**External Web Tools**» пункт «**$(PATH)**» поднять выше пунктов «**$(VSINSTALLDIR)**» (см. рис. Ниже);



1. Так же в системе должен быть установлен «**TypeScript 2.2 for Visual Studio 2015**» (или выше) – <https://www.microsoft.com/en-us/download/details.aspx?id=48593>. После установки необходимо перезагрузить компьютер.

Подробнее о проектах с angular 4 можно узнать тут – <https://angular.io/guide/visual-studio-2015> .

### Сборка

После разворачивания проекта из репозитория необходимо подгрузить все пакеты и для **ASP.NET** (подгрузятся автоматом) и для **Angular**. Загрузку пакетов для Angular (если не загрузятся автоматически) можно запустить вручную, для этого необходимо правой кнопкой мыши нажать над файлом «**package.json**» и в выпадающем меню выбрать пункт «**восстановить пакеты**». Если все сделано правильно и выполнены все требования (см. пред. пункт), то в папке основного проекта будет создана папка «**node\_modules**» (в состав проекта она не входит, в VS ее не видно).

## Работа программы

Программа состоит из двух форм – «Список книг» и «Редактирование книги».

|  |  |
| --- | --- |
| Рисунок 1. Список книг | Рисунок 2. Редактирование |

**В программе реализованы все требования**:

1. Принципы **KIS/DRY**;
2. Использование технологий **asp.net mvc**, **web api 2**, **js/jquery;**
3. Программе представляет собой **SPA-приложение**, которое построено с помощью SPA-фреймворка «**Angular 4**»;
4. Хранение данных в памяти сервера. Проект asp.net создает статический класс-репозиторий, которые хранит данные;
5. Оформление в виде «**Grid**»;
6. В программе реализована **Валидация** и на клиенте и на сервере. **На клиенте** при создании/редактировании книги (см. рис.2) все поля валидируются согласно заданию. Если хотя бы одно поле не соответствует требованиям, то кнопка «Сохранить» будет не активна и под соот. полем будет информация об ошибке. **На сервере** – в контроллерах идет валидация присланной для вставки или редактирования модели стандартными средствами ASP.NET (атрибуты валидации прописаны в классе модели).
7. Использование DI – в проекте используется «**Ninject**».
8. Программа позволяет добавлять, удалять и редактировать существующие книги и ее авторов;
9. Реализована **возможность** **сортировки** по заголовку и году публикации. **Сортировка сохраняется после перезагрузки страницы**.
10. Реализована **возможность загрузки изображения**.

**Упрощения**:

1. В программе можно было бы еще сделать справочник издателей, чтобы бд соответствовала 3й нормальной форме. Т.к. задание тестовое, то это опущено для простоты;
2. Для справочника авторов реализована только возможность вставки. Можно сделать полноценный справочник в будущем.
3. Валидация ISBN реализована только для 10знакового формата (поздно узнал, что есть еще и 13 знаковый формат).
4. Не реализовано корректное масштабирование изображений.